



Badanie strategii rozmieszczenia elementów objektowych baz danych

JAROSŁAW KOSZELA, TADEUSZ NOWICKI

Wojskowa Akademia Techniczna, Wydział Cybernetyki, Instytut Systemów Informatycznych,
00-908 Warszawa, ul. S. Kaliskiego 2

Streszczenie. W pracy przedstawiono metodę badania wydajności rozproszonej obiektowej bazy danych, a w szczególności jednego z podstawowych aspektów związanych ze zwiększaniem wydajności rozproszonego systemu — alokacji jej elementów. Przetawiono definicję modelu klas, obiektów, atrybutów, metod i powiązań w rozproszonej obiektowej bazie danych. Zdefiniowano model zapytania, strukturę systemu rozproszonego, model rozproszenia zasobów obiektowej bazy danych oraz oszacowano jednostkowy czas odpowiedzi systemu i koszt wykonania zapytania przez dany węzeł, jak i cały system rozproszony. Sformułowano zadanie wyznaczania zasobów obiektowej bazy danych. Praca opisuje metodę wyznaczania optymalnego rozmieszczenia atrybutów, metod i powiązań w rozproszonej obiektowej bazie danych, wykorzystując algorytmy genetyczne. Zaprezentowano zastosowane metody generowania kolejnych elementów populacji (alokacji zasobów) dla wyznaczenia rozwiązania optymalnego. Przedstawiono aplikację do wyznaczania rozmieszczenia elementów obiektowej bazy danych. Pokazano też wybrane własności uzyskanego rozwiązania.

Słowa kluczowe: modelowanie systemów informatycznych, obiektowe bazy danych, rozproszone bazy danych, alokacja elementów baz danych

Symbole UKD: 681.3.05

1. Wprowadzenie

Decyzja o budowie systemu informatycznego w postaci systemu rozproszonego uwarunkowana jest określonymi wymaganiami, jakie im się stawia. Wymagania te dotyczą głównie trzech aspektów: wydajności, bezpieczeństwa oraz aspektu ekonomicznego (kosztowego). Każdy z tych aspektów niesie potencjalne korzyści, jak i problemy, które trzeba rozważyć w przypadku podjęcia decyzji o budowie systemu tej klasy.

Systemy przetwarzania rozproszonego funkcjonują na zasadzie otwartej komunikacji międzyprocesowej. Wykorzystując standardy systemów otwartych, można zbudować skalowalne środowisko w zakresie sprzętu, oprogramowania oraz liczby jednocześnie obsługiwanych użytkowników. Z tego względu systemy te są również nazywane otwartymi systemami rozproszonymi. Znajdują one powszechne zastosowanie nie tylko w projektach badawczych, ale przede wszystkim we wdrożeniach komercyjnych.

Podstawowym elementem, wręcz fundamentem niemalże każdego eksploatowanego czy tworzonego współczesnego systemu informatycznego, są zbiory danych. Często zasoby danych przechowywane w ramach danego systemu informatycznego są bardzo duże, co wymusza tworzenie narzędzi zarządzania i operowania na tych danych.

Geneza obiektowych baz danych leży u podstaw obiektowych i zorientowanych obiektowo języków programowania. Metody i języki obiektowe zdobyły bardzo dużą popularność i stały się podstawowymi narzędziami budowy większości systemów informatycznych. W tym samym czasie największą popularność na rynku miały natomiast systemy relacyjnych baz danych, które stanowią bardzo istotny składnik systemów informatycznych.

Model obiektowy, który precyzuje znaczenie podstawowych pojęć obiektowości, takich jak: obiekty, klasy, metody, dziedziczenie i hermetyzacja, ustala również podstawowe założenia dotyczące struktur danych przechowywanych przez system. Takimi strukturami są: obiekty, atrybuty i związki (*relationships*). Klasy obiektów podlegają dziedziczeniu i hermetyzacji. Model obiektowy wprowadza pojęcie identyfikatora obiektu (OID — *object identifier*), jako unikalnego w całym systemie, niezmiennego i niezależnego od wartości atrybutów obiektu oraz automatycznie generowanego przez system mechanizmu określającego tożsamość obiektu. Wprowadza także pojęcie literału, czyli takiego obiektu, którego nie można modyfikować. Istotnym założeniem modelu obiektowego ODMG jest mocna kontrola typów oraz pełna dowolność w zakresie kombinacji konstruktorów typów, w tym konstruktorów typów masowych (kolekcji), takich jak: zbiór, wielozbiór, sekwencja i tablica dynamiczna.

Wraz z pojęciem rozproszonych baz danych koegzystuje pojęcie rozproszonego systemu zarządzania bazą danych (RSZBD), jako jednej logicznej bazy danych podzielonej na fragmenty. Każdy fragment przechowywany jest na jednym bądź większej liczbie komputerów i zarządzany jest przez osobny system zarządzania bazą danych. Komputery (węzły), zawierające poszczególne fragmenty bazy, połączone są poprzez sieć komputerową. Każdy komputer może niezależnie przetwarzać żądania użytkowników, dotyczące jego lokalnych danych (czyli każda jednostka systemu posiada pewien zakres autonomii), a także przetwarzać dane przechowywane na innych komputerach w sieci.

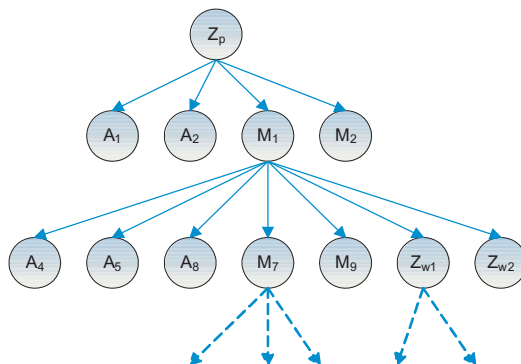
Oprogramowanie umożliwia połączenie rozproszonych zasobów w jedną całość, utrzymanie spójności zasobów oraz udostępnianie ich użytkownikom, przy założeniu przezroczystości rozproszenia.

Jednym z ważniejszych problemów, stojących przed projektantem obiektowych baz danych, jest alokacja obiektów w węzłach rozproszonego systemu komputerowego. Alokacja jest procesem przydziału każdego fragmentu do jednego lub więcej miejsc (węzłów). Podstawowym celem alokacji jest zredukowanie kosztów przetwarzania. Alokacja rozpatrywana jest w kontekście dwóch miar: minimalizacji kosztu (koszt przechowywania każdego fragmentu w węźle sieci, koszt ewaluacji zapytania do fragmentu w węźle, koszt modyfikacji jednego fragmentu we wszystkich węzłach systemu rozproszonego, gdzie jest on przechowywany) oraz kosztu transmisji. Podstawowym problemem alokacji jest znalezienie przydziału, który minimalizuje łączną funkcję kosztów oraz wydajności — odpowiednio zdefiniowana strategia alokacji powinna utrzymać wydajność systemu na żądanym poziomie i odnosić się głównie do minimalizacji czasu odpowiedzi systemu i maksymalizacji przepustowości systemu dla każdego węzła rozproszonego systemu.

W procesie alokacji fragmentów zbiorów bazy danych bardzo istotnymi aspektami, które należy uwzględnić, są wymagania związane z funkcjonowaniem rozpatrywanego rozproszonego systemu, do których zalicza się: dane ilościowe o bazie danych (pojemność, liczba elementów), sieci komputerowej (przepustowość, topologia), możliwościach wydajnościowych poszczególnych węzłów rozproszonego systemu (moc obliczeniowa, pojemność pamięciowa) i ograniczeniach składowania w każdym węźle systemu oraz liczbę typów zapytań wraz z liczbą zapytań danego typu.

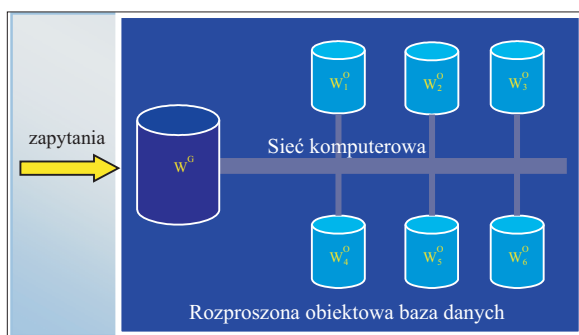
Podstawowymi aspektami związanymi z wydajnością pracy rozproszonego systemu baz danych są: odpowiednio skonstruowana fragmentacja zasobów, ich właściwa alokacja w poszczególnych węzłach rozproszonego systemu oraz metody optymalizujące realizacje, napływających do systemu zapytań klientów. Odpowiednia alokacja powinna uwzględniać zarówno architekturę systemu ROBD (ograniczenia wydajnościowe i pojemnościowe węzłów systemu, sieć komputerową i wykorzystane oprogramowanie), schemat ROBD (definicje klas, powiązań), jak i uwzględniać aspekty związane z bezpieczeństwem, niezawodnością i wymaganiami biznesowymi. Dlatego też bardzo istotnym problemem jest opracowanie metody rozmieszczenia początkowego zasobów ROBD, jak też możliwość rozmieszczenia zasobów ROBD już eksploatowanej (np. w trakcie prac serwisowych), która pozwoliłaby na efektywniejsze działanie ROBD, z punktu widzenia obsługi napływających od klientów zapytań (skrócenie czasu realizacji zapytania, jak również zmniejszenie kosztu wykonania zapytania). Należy zwrócić uwagę na fakt, że w obecnie stosowanych rozwiązaniach nie uwzględnia się możliwości powstania kaskady zapytań wtórnych wewnątrz ROBD, powstałej w wyniku realizacji zapytania otrzymanego od klienta (zapytanie pierwotne).

Rozproszona obiektowa baza danych obsługuje napływające od klientów zapytania (zapytania pierwotne), które system musi w sposób jak najbardziej wydajny przetworzyć i zwrócić wyniki tych zapytań. Wykorzystanie w strukturze zapytania (poza atrybutami) również metod, powoduje możliwość wystąpienia zjawiska kaskady zapytań wtórnych, które muszą być wykonane w celu poprawnej realizacji zapytania pierwotnego (rys. 1).



Rys. 1. Przykład powstawania kaskady zapytań w bazie obiektowej

System obiektowych baz danych działa w rozproszonym środowisku heterogenicznym o skończonej liczbie węzłów (rys. 2). Rozproszenie ma charakter „lokalny”, tzn. parametry sieci (czas odpowiedzi, przepustowość itp. — np. 1 Gb lub 10 Gb Ethernet) nie powinny być „wąskim gardłem” dla systemu. W systemie określone są węzły operacyjne, których podstawowym zadaniem jest gromadzenie i przetwarzanie zgromadzonych zasobów i węzeł (węzły) główny, jako fizyczny lub logiczny element systemu odpowiedzialny za gromadzenie informacji, związanych ze schematem bazy obiektowej, rozproszeniem zasobów bazy obiektowej po węzłach operacyjnych (może to być wydzielony węzeł lub węzły systemu, jak również wydzielone zasoby węzłów operacyjnych). Węzły sieci są niejednorodne. Niejednorodność węzłów wynika



Rys. 2. Schemat rozproszonej obiektowej bazy danych

z różnorodności wartości parametrów określających zasoby tych węzłów (np. moc obliczeniowa, pamięć zewnętrzna, maksymalna przepustowość medium transmisyjnego itp.). Badany system wykorzystuje sieć komputerową typu Ethernet.

2. Optymalizacja rozmieszczania elementów obiektowej bazy danych

W ramach opracowanej metody rozpraszania elementów obiektowej bazy danych należy zdefiniować i określić następujące elementy:

- model architektury badanego systemu rozproszonego, definicje parametrów ilościowych, wydajnościowych i pojemnościowych węzłów systemu,
- formalny model klas obiektów, który określa strukturę danych gromadzoną w systemie obiektowych baz danych wraz z definicją atrybutów i metod przypisanych do klas obiektów,
- możliwość wywołania metody przez inną (wysłanie komunikatu),
- formalny model powiązań występujący w modelu struktury klas wraz z definicjami powiązań typu dziedziczenie, agregacja, kompozycja i asocjacja, które definiowane są w typowym modelu klas,
- formalny model obiektów pozwalający na określenie przynależności obiektów do klas, liczności zbioru obiektów danej klasy (ekstensja), powiązania pomiędzy obiektami, ich rodzaj i licznosc.

Model rozproszonej bazy danych odnosi się do rozmieszczenia atrybutów, metod i powiązań w rozproszonym systemie komputerowym. Rozmieszczenie atrybutów definiuje się przez macierz:

$$B^A = [b_{a,p}^A]_{A \times P^A}, \quad (1)$$

gdzie $b_{a,p}^A$ jest równe 1, gdy atrybut a jest przechowywany w p -tej paczce i 0 w przeciwnym przypadku, przy ograniczeniach:

- atrybut może być przypisany do jednej paczki

$$\forall a \in A; \sum_{p=1}^{P^A} b_{a,p}^A = 1 \quad (2)$$

- paczka zawiera co najmniej jeden atrybut

$$\forall p \in P^A; \sum_{a=1}^A b_{a,p}^A \geq 1 \quad (3)$$

Rozmieszczenie metod definiuje się przez macierz:

$$B^M = [b_{m,p}^M]_{M \times P^M}, \quad (4)$$

gdzie $b_{a,p}^M$ jest równe 1, gdy metoda m jest przechowywana w p -tej paczce i 0 w przeciwnym przypadku, przy ograniczeniach:

- metoda może być przypisana do minimum jednej paczki

$$\forall m \in M; \sum_{p=1}^{P^M} b_{m,p}^M \geq 1 \quad (5)$$

- paczka zawiera co najmniej jedną metodę

$$\forall p \in P^M; \sum_{m=1}^M b_{m,p}^M \geq 1 \quad (6)$$

Rozmieszczenie powiązań definiuje się przez macierz:

$$B^R = [b_{r,p}^R]_{R \times P^R}, \quad (7)$$

gdzie $b_{a,p}^R$ jest równe 1, gdy powiązanie r -te jest przechowywane w p -tej paczce i 0 w przeciwnym przypadku, przy ograniczeniach:

- powiązanie może być przypisane do jednej paczki

$$\forall r \in R; \sum_{p=1}^{P^R} b_{r,p}^R = 1 \quad (8)$$

- paczka zawiera co najmniej jedno powiązanie

$$\forall p \in P^R; \sum_{r=1}^R b_{r,p}^R \geq 1. \quad (9)$$

Zmienną decyzyjną w problemie rozmieszczenia atrybutów, metod i powiązań jest trójka uporządkowana:

$$X = \langle X^A, X^M, X^R \rangle, \quad (10)$$

gdzie alokacja paczek atrybutów w poszczególnych węzłach określa macierz:

$$X^A = [x_{w,p}^A]_{W \times P^A}, \quad (11)$$

alokacja paczek metod w poszczególnych węzłach określa macierz:

$$X^M = [x_{w,p}^M]_{W \times P^M}, \quad (12)$$

alokacja paczek powiązań w poszczególnych węzłach określa macierz:

$$X^R = [x_{w,p}^R]_{W \times P^R}. \quad (13)$$

Zakłada się, że mamy do czynienia z N typami podstawowych zapytań do bazy danych, a definicja zapytania jest następująca:

$$Q = \langle Q^1, Q^2, Q^3, Q^4 \rangle, \quad (14)$$

gdzie: Q^1 — oznacza wektor określający uwzględnione klasy w zapytaniu Q ,
 Q^2 — oznacza wektor określający bezpośrednio uwzględnione atrybuty w zapytaniu Q ,
 Q^3 — oznacza wektor określający bezpośrednio uwzględnione metody w zapytaniu Q ,
 Q^4 — oznacza wektor określający bezpośrednio uwzględnione powiązania w tym zapytaniu.

Zapytania mogą wywoływać kaskadę zapytań wtórnych o metody, a te wywoływać mogą kolejne zapytania. Powoduje to konieczność rozpatrzenia rekurencyjnych równań dla zapytań do obiektowej bazy danych, dla jednostkowych czasów funkcjonowania tej bazy. Dla przyjętych założeń w rozpatrywanym modelu można przyjąć, że gdy oznaczmy przez:

$N_M^m(1)$ — jednostkowa liczba metod typu m (pierwotnych i wtórnych),

$N_Z^z(1)$ — jednostkowa liczba zapytań typu z (pierwotnych i wtórnych),

to układ równań stochastycznych określających związki między tymi wielkościami ma postać:

$$N_M^m(1) = \sum_{z=1}^Z N_Z^z(1) \cdot I^M(z, m) + \sum_{\substack{n=1 \\ n \neq m}}^M N_M^n(1) \cdot p_{n,m}^M, \quad m = \overline{1, M} \quad (15)$$

$$N_Z^z(1) = N_{Z_p}^z(1) + \sum_{m=1}^M N_M^m(1) \cdot p_{m,z}^Z, \quad z = \overline{1, Z}.$$

Chcąc wyznaczyć charakterystyki w sensie wartości oczekiwanych i przyjmując następujące oznaczenia:

$H_Z^z(1) = E \{N_Z^z(1)\}$ — oczekiwana liczba zapytań pierwotnych i wtórnych typu z dla jednostkowego czasu funkcjonowania ROBD,

$H_M^m(1) = E \{N_M^m(1)\}$ — oczekiwana liczba metod pierwotnych i wtórnych typu m dla jednostkowego czasu funkcjonowania ROBD,

to układ równań w sensie wartości oczekiwanych do wyznaczenia powyższych wielkości jest następujący:

$$\begin{cases} H_M^m(1) = \sum_{z=1}^Z H_Z^z(1) \cdot I^M(z, m) + \sum_{\substack{n=1 \\ n \neq m}}^M H_M^n(1) \cdot e_{n,m}^M, & m = \overline{1, M} \\ H_Z^z(1) = H_{Zp}^z(1) + \sum_{m=1}^M H_M^m(1) \cdot e_{m,z}^Z, & z = \overline{1, Z}. \end{cases} \quad (16)$$

Oszacowanie kosztu obciążenia węzła przetwarzanymi zapytaniem jest sumą kosztów związanych z przetworzeniem i przesłaniem wybranych w zapytaniach zasobów systemu (atrybutów obiektów klas i powiązań) oraz wykonania wybranych w zapytaniach metod w danym węźle:

$$\begin{aligned} K_{jedn}^w(X^A, X^M, X^R) = & \frac{1}{\beta_w} \left(\sum_{z=1}^Z w_z \cdot H^z(1) \sum_{c=1}^C Q_z^1(c) \cdot T(X^A, X^R, w, c, z) \right) + \\ & + \frac{1}{\beta_w} \left(\sum_{z=1}^Z \sum_{m=1}^M w_z \cdot H^m(1) \sum_{p=1}^{p^M} Q_z^3(m) \cdot b_{m,p}^M \cdot x_{w,p}^M \cdot T_2(m) \right). \end{aligned} \quad (17)$$

Pierwsza część powyższego wzoru dotyczy kosztów związanych z obciążeniem węzła zapytaniem, a druga — obciążeniem węzłów metodami. Oszacowanie kosztu obciążenia węzła przetwarzanymi zapytaniem wynosi:

$$\begin{aligned} T(X^A, X^R, w, z, c, r) = & S^A(X^A, z, c, w) + \\ & + \sum_{r=1}^R Q_z^4(r) \cdot \left\{ \left(\sum_{c_1=c+1}^C Q_z^1(c_1) \cdot I(c, c_1, r) \right) \cdot \right. \\ & \cdot \left(\sum_{p=1}^{p^R} b_{r,p}^R \cdot x_{w,p}^R \cdot T_3(Poj^R(r)) \right) + \\ & + [I(c, c, r) \cdot S^R(X^A, r, c, w, z)] - \\ & + \left[(1 - \bar{S}(c_1)) \cdot S^A(X^A, z, c, w) \right] \cdot \\ & \cdot \left[1 - \sum_{c_1=1}^C Q_z^1(c_1) \cdot I(c, c_1, r) \right], \end{aligned} \quad (18)$$

gdzie składniki powyższej formuły są następujące:

$$S^A(X^A, z, c, w) = \sum_{a=1}^A \left[Q_z^2(a) \cdot I^A(c, a) \cdot \bar{Q}_z^2(a) \cdot \sum_{p=1}^{p^A} b_{a,p}^A \cdot x_{w,p}^A \cdot T_1(Poj^A(a)) \right] \quad (19)$$

$$\begin{aligned}
S^R(X^R, r, c, w, z) &= \sum_{p=1}^{P^R} b_{r,p}^R \cdot x_{w,p}^R \cdot T_3(Poj^R(r)) + \\
&+ \text{sign} \left(\sum_{a=1}^A \sum_{p=1}^{P^A} Q_z^2(a) \cdot I^A(c, a) \cdot b_{a,p}^A \cdot x_{w,p}^A \right) \cdot \\
&\cdot \sum_{\substack{w_i=1 \\ w_i \neq w}}^W \sum_{p=1}^{P^R} b_{r,p}^R \cdot x_{w_i,p}^R \cdot T_3(Poj^R(r))
\end{aligned} \tag{20}$$

Wtedy zadanie optymalizacji — minimalizacja kosztu obsługi zapytań do ROBD, jest minimalizacją oczekiwanego czasu odpowiedzi obiektowej bazy danych na zapytania:

$$\bar{T}^*(X^*) = \min_{x \in \Omega} \bar{T}(X), \tag{21}$$

przy ograniczeniach podstawowych (pojemnościowych węzłów sieci komputerowej)

$$\begin{aligned}
\forall i \in W; \quad &\sum_{p=1}^{P^A} \sum_{k=1}^A (x_{i,p}^A \cdot b_{k,p}^A \cdot Poj^A(a_k)) + \\
&+ \sum_{p=1}^{P^M} \sum_{k=1}^M (x_{i,p}^M \cdot b_{k,p}^M \cdot Poj^M(m_k)) + \\
&+ \sum_{p=1}^{P^R} \sum_{k=1}^R (x_{i,p}^R \cdot b_{k,p}^R \cdot Poj^R) \leq w_i^H
\end{aligned} \tag{22}$$

oraz innych o charakterze formalnym. Można jednak przyjąć, że równoważenie obciążenia (*load balancing*) we wszystkich węzłach sieci komputerowej w konsekwencji powinno prowadzić do minimalizacji czasu odpowiedzi (reakcji) ROBD. Jest to równoważne z minimalizacją poniższej formuły:

$$\begin{aligned}
&\sum_{w=1}^W [K^w(X^A, X^M, X^R) - \bar{K}]^2 = \\
&= \frac{1}{W^2} \sum_{w=1}^W \left[\left[W \cdot K^w(X^A, X^M, X^R) - \sum_{i=1}^W K^i(X^A, X^M, X^R) \right]^2 \right].
\end{aligned} \tag{23}$$

3. Rozmieszczenie zasobów rozproszonej obiektowej bazy danych

Podczas konstruowania baz danych w systemach rozproszonych jednym z istotnych, z projektowego punktu widzenia, jest problem wyznaczenia początkowego rozmieszczenia zasobów bazy danych w węzłach komputerowego systemu rozproszonego. Problemy tego typu dla klasycznych baz danych, w tym również dla relacyjnych baz danych, były formułowane i rozwiązywane. Od właściwego rozmieszczenia zasobów baz danych zależy bardzo podstawowy parametr efektywnościowy, zwany czasem odpowiedzi (ang. *response time*). Dąży się do tego, aby miał on jak najmniejszą wartość. Można pokazać [pkt 2], że w przypadku obiektowych baz danych problem efektywnego rozmieszczenia zasobów jest szczególnie złożony,

X^A	Pa_1	Pa_2	...	Pa_P^A
w_1	$X_{1,1}^A$	$X_{1,2}^A$...	$X_{1,P}^A$
w_2	$X_{2,1}^A$	$X_{2,2}^A$...	$X_{2,P}^A$
...
w_w	$X_{w,1}^A$	$X_{w,2}^A$...	$X_{w,P}^A$
X^M	Pm_1	Pm_2	...	Pm_P^M
w_1	$X_{1,1}^M$	$X_{1,2}^M$...	$X_{1,P}^M$
w_2	$X_{2,1}^M$	$X_{2,2}^M$...	$X_{2,P}^M$
...
w_w	$X_{w,1}^M$	$X_{w,2}^M$...	$X_{w,P}^M$
X^R	Pr_1	Pr_2	...	Pr_P^R
w_1	$X_{1,1}^R$	$X_{1,2}^R$...	$X_{1,P}^R$
w_2	$X_{2,1}^R$	$X_{2,2}^R$...	$X_{2,P}^R$
...
w_w	$X_{w,1}^R$	$X_{w,2}^R$...	$X_{w,P}^R$

Rys. 3. Macierze rozmieszczenia atrybutów, metod i powiązań w obiektowej bazie danych

ponieważ zapytania do baz danych powodują całą kaskadę dodatkowych odwołań do metod umieszczonych w obiektach, a nawet generowana jest kaskada wtórnych zapytań do bazy danych.

Atrybuty, metody i powiązania obiektów w rozproszonej obiektowej bazie danych (ROBD) są rozmieszczone w trzech macierzach. Macierze te, zdefiniowane w pkt. 2, mają postać, jak pokazano na rysunku 3.

W pkt. 2 pokazano konstrukcję kryterium poszukiwania rozmieszczenia (alokacji) zasobów obiektowej bazy danych. Można zatem sformułować zadanie poszukiwania najlepszego rozmieszczenia zasobów minimalizującego wyznaczone kryterium.

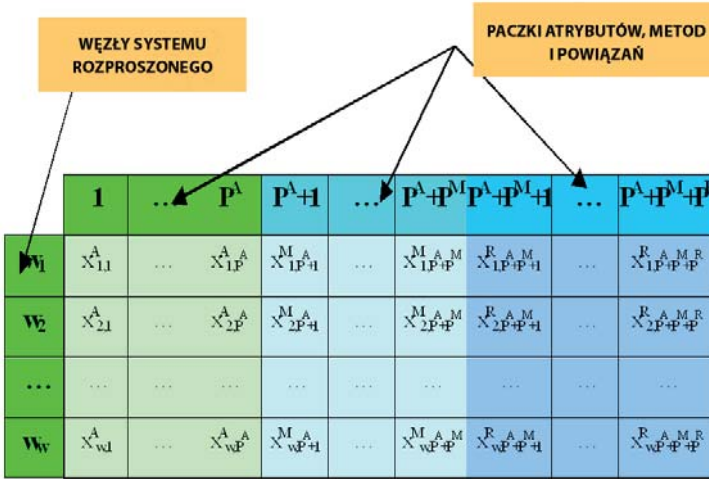
4. Metoda optymalizacji rozmieszczenia zasobów rozproszonej obiektowej bazy danych

Chcąc poszukiwać najlepszego rozmieszczenia atrybutów, metod i powiązań w rozproszonej obiektowej bazie danych, należy rozwiązać zadanie optymalizacji z punktu widzenia kryterium przedstawionego w pkt. 2. Jest to kryterium nieliniowe (w pewnych fragmentach wyznaczane algorytmicznie), zatem klasyczne metody wyznaczania rozwiązania optymalnego nie mogą być stosowane.

Powstała zatem, naturalna w tym przypadku, koncepcja skorzystania z metod stochastycznego poszukiwania ekstremum funkcji. W grupie tych metod niezwykle popularnymi jawią się metody zwane algorytmami genetycznymi. Polegają one, w dużym przybliżeniu, na generowaniu kolejnych, coraz lepszych rozwiązań na podstawie już posiadanych. Oczywiście kolejnym problemem staje się wyznaczenie populacji początkowej (bazowej) rozwiązań. Ponadto należy zdefiniować mechanizm generowania kolejnych rozwiązań (potomków) na podstawie istniejącej populacji rozwiązań (rodziców). Aby zdefiniować algorytm genetyczny, rozwiązujący problem wyznaczania optymalnego rozmieszczenia zasobów obiektowej bazy danych, trzeba zdefiniować tak zwany chromosom (sposób zapisu, model) rozwiązania problemu, czyli postać osobnika w populacji rozwiązań. Chromosom taki zdefiniowany został przez połączenie trzech macierzy, co przedstawiono symbolicznie na rysunku 4.

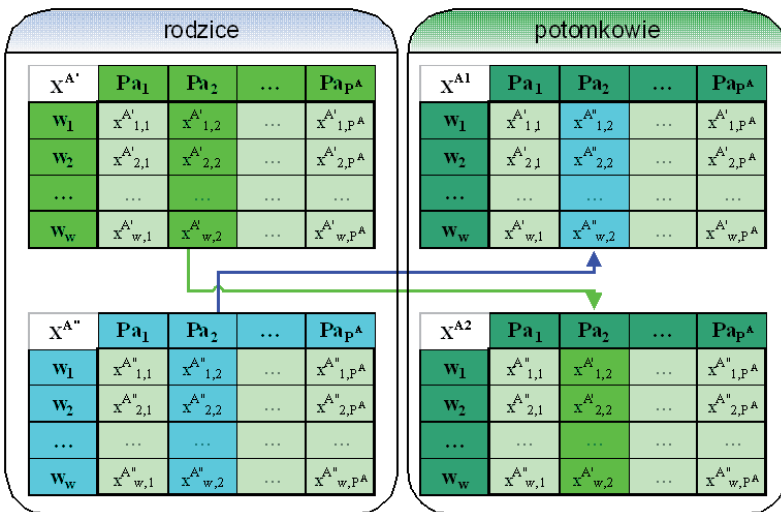
W kolejnym kroku można zdefiniować sposoby tworzenia kolejnych dopuszczalnych rozwiązań problemu rozmieszczenia zasobów obiektowej bazy danych na podstawie istniejących już rozwiązań. W pracy wykorzystano kilka metod tak zwanego krzyżowania istniejących rozwiązań i metod mutacji rozwiązań.

Operacja krzyżowania polega na tym, że na podstawie dwóch osobników powstaje nowy osobnik o cechach różnych od rodziców. Krzyżowanie, wykorzystane w tej metodzie, zostało zaimplementowane w postaci kilku wariantów tego mechanizmu: krzyżowanie typu „MixCol”, krzyżowanie typu „MixColPos”, krzyżowanie typu „MixNCol”, krzyżowanie typu „MixColAlfa”, krzyżowanie typu „MixNColAlfa” oraz krzyżowanie z użyciem parametru α „MixAlfa”.



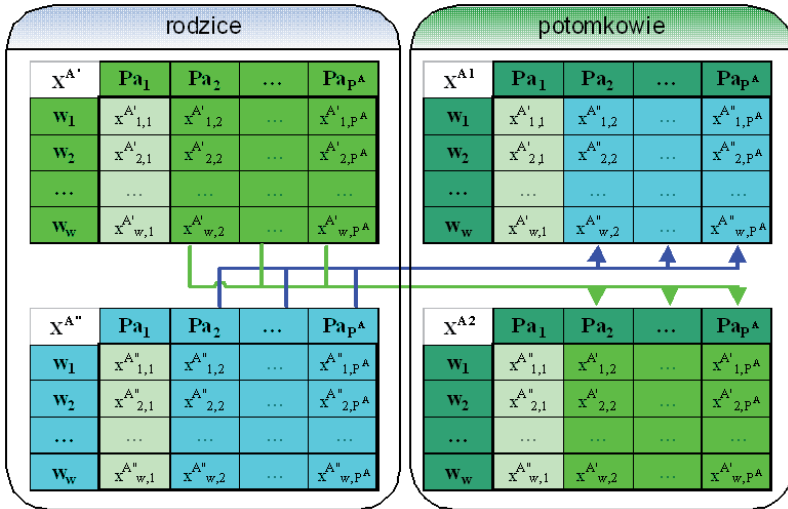
Rys. 4. Macierze rozmieszczenia atrybutów, metod i powiązań w obiektowej bazie danych

W operacji MixCol losowo wskazywana kolumna (gen) każdego z rodziców jest wymieniana, tworząc potomka. Faktycznie w tym przypadku bierze się pod uwagę dwóch rodziców, z których wymienia się losowo wskazaną kolumnę chromosomu, przy czym dokonuje się przy tym weryfikacji i ewentualnej korekcji rozwiązania, aby spełniło ono ograniczenia dla dopuszczalnego rozwiązania problemu rozmieszczenia zasobów obiektowej bazy danych. Operacja MixCol została pokazana symbolicznie na rysunku 5. Warto pamiętać, że w przypadku tej operacji zamienia się tylko jedną z kolumn w chromosomie definiującym rodzica.

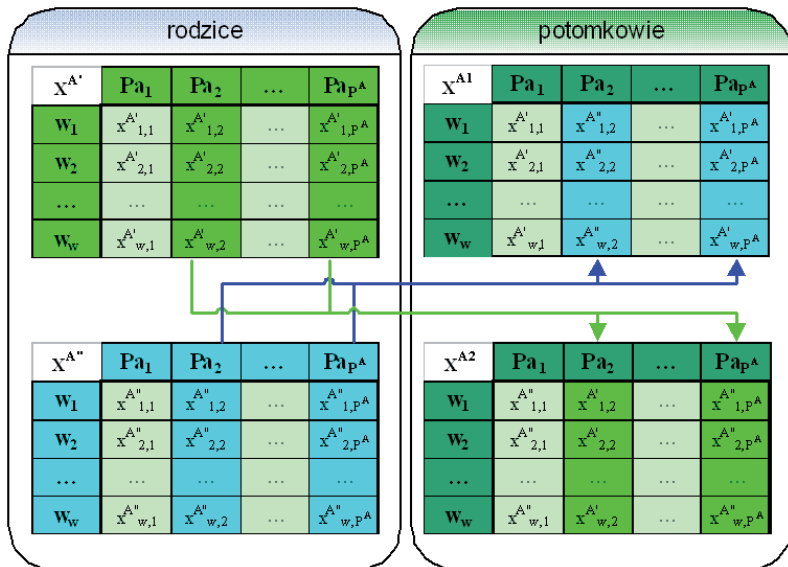


Rys. 5. Operacja MixCol generowania potomków

Kolejna operacja MixColPos różni się od MixCol tym, że wskazywane są grupy sąsiadujących ze sobą kolumn, które zamienia się u rodziców, tworząc kolejnych potomków (rys. 6).



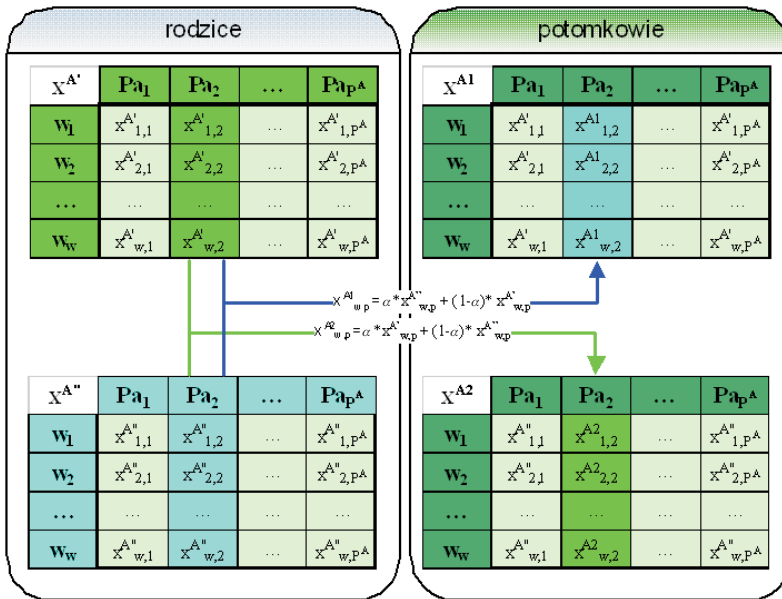
Rys. 6. Operacja MixColPos generowania potomków



Rys. 7. Operacja MixNCol generowania potomków

Rozwinięcie poprzednich operacji krzyżowania to operacja MixNPos, która różni się od MixCol tym, że wskazywane są losowo liczne kolumny, niekoniecznie sąsiednie, które zamienia się u rodziców, tworząc kolejnych potomków (rys. 7).

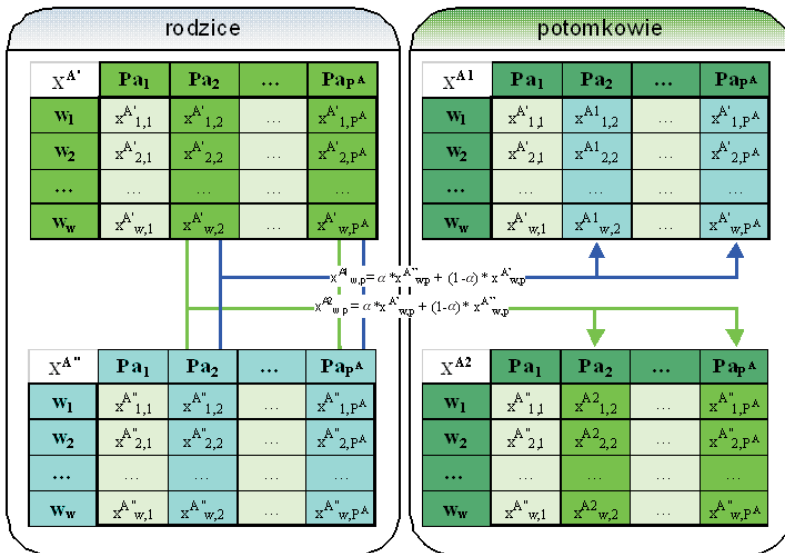
Kolejną modyfikacją metody krzyżowania jest operacja MixColAlfa, w której losowo wybrana, pojedyncza kolumna u jednego z rodziców mnożona jest przez wartość α , natomiast ta sama kolumna u drugiego z rodziców mnożona jest przez liczbę $(1-\alpha)$ i odwrotnie, przy czym $\alpha \in (0,1)$. Kombinacje wypukłe kolumn tworzą nowych potomków (rys. 8).



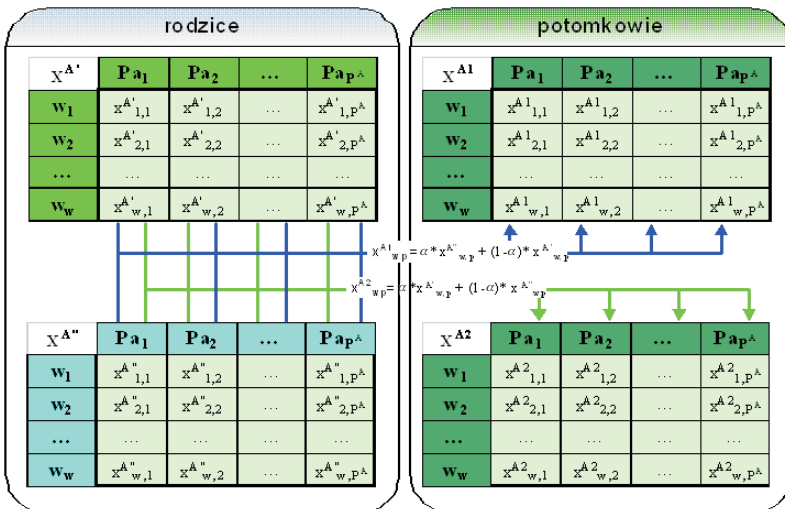
Rys. 8. Operacja MixColalfa generowania potomków

Pierwsza modyfikacja metody MixColAlfa, nazwana MixNColAlfa, polega na tym, że losowo wskazane kolumny, a nie pojedyncze, są mnożone przez parametry α i $(1-\alpha)$, a stosowne kombinacje wypukłe tworzą nowych potomków (rys. 9).

Ostatnia z metod generowania potomków populacji rozwiązań, definiujących rozmieszczenie zasobów w obiektowej bazie danych, polega na wyborze nie losowej liczby kolumn. Wszystkie kolumny podlegają mnożeniu przez odpowiednie parametry i tworząc kombinacje wypukłe, są wstawiane do tworzonych macierzy, tworząc kolejnych potomków dla rozpatrywanej populacji. Metoda ta nazwana została krótko MixAlfa i jest najbogatszym rozszerzeniem operacji krzyżowania, w których dokonuje się generowanie potomków przez tworzenie kombinacji wypukłych kolumn z chromosomów. Schemat operacji MixAlfa został pokazany symbolicznie na rysunku 10.



Rys. 9. Operacja MixNColAlfa generowania potomków

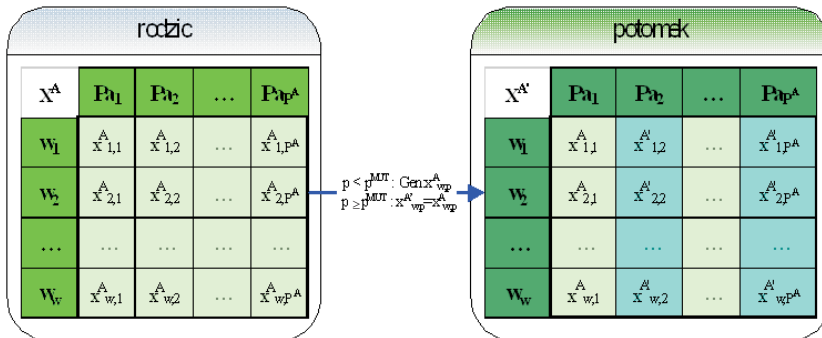


Rys. 10. Operacja MixAlfa generowania potomków

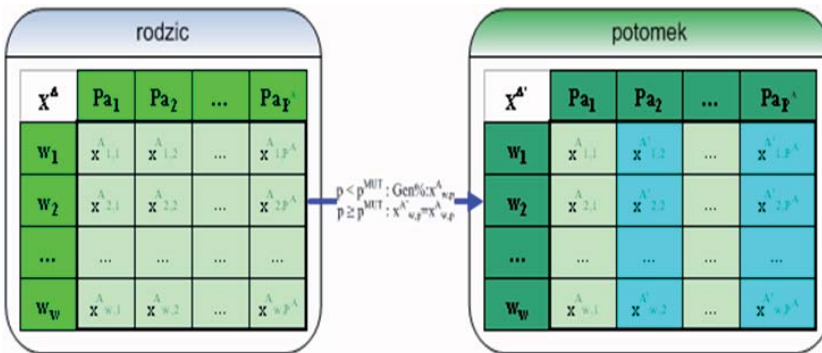
Z kolei operacje mutacji powodują, że z jednego osobnika powstaje nowy osobnik, którego wybrane cechy mogą być modyfikowane — również może być różny od rodzica. Pierwszą z operacji mutowania jest operacja „grube-sito”, w której losowo wskazane kolumny (geny) rodzica są modyfikowane — ponownie generowane tak, aby spełniały ograniczenia, przy czym nie ma na to wpływu

poprzednia postać kolumny, tworząc potomka. Operacja ta została przedstawiona symbolicznie na rysunku 11.

Drugą z operacji mutowania jest operacja „drobne-sito”, w której losowo wskazana kolumna (gen) rodzica jest modyfikowana — ponownie generowana tak, aby spełniała ograniczenia, przy czym teraz ma wpływ poprzednia postać kolumny, ponieważ mnożymy jej wartości przez pewien ułamek, tworząc potomka. Operacja ta została przedstawiona symbolicznie na rysunku 12.



Rys. 11. Operacja „grube-sito” generowania potomków



Rys. 12. Operacja „drobne-sito” generowania potomków

5. Implementacja algorytmu genetycznego

Środowisko badawcze użyte do badania własności rozproszonych obiektowych baz danych jest oprogramowaniem, które wykorzystując opracowane modele matematyczne funkcjonowania tej klasy systemów i metodę optymalizacji, wykorzystującą algorytmy genetyczne, pozwala na uzyskanie wyników, które powinny zwiększyć wydajność funkcjonowania tej klasy systemów. Opracowane

środowisko ma charakter eksperymentalny, ponieważ nie są dostępne rzeczywiste systemy tej klasy, na których można byłoby dokonać weryfikacji prezentowanych rozwiązań. Poniżej pokazano ekran prezentujący interfejs do wprowadzania i modyfikacji parametrów konfiguracyjnych środowiska rozproszonego systemu komputerowego oraz parametry strumieni zapytań do bazy danych.

The screenshot shows the ROBD configuration window with the following data:

Parametry:

- Liczba klas (C): 5
- Liczba węzłów op (W): 5
- Liczba typów zapytań (Z): 5
- Liczba atrybutów (A): 10
- Liczba metod (M): 10
- Liczba powiązań (R): 5
- Liczba typów paczek atrybutów (Pa): 5
- Liczba typów paczek metod (Pm): 5
- Liczba typów paczek powiązań (Pr): 5
- Poj[ref] [B]: 16
- Poj OID [B]: 16

Liczba obiekt. klas:

1:1	C1	C2	C3	C4	C5
LOC	10000000	20000000	30000000	40000000	50000000

Zajętość pamięci przez atrybuty [B]:

1:1	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
AttrMem	1	2	3	4	5	6	7	8	9	10

Zajętość pamięci przez metody [B]:

1:1	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10
MetMem	10	9	8	7	6	5	4	3	2	1

Wielkość dostępnej pamięci w węzle [GB]:

1:1	w1	w2	w3	w4	w5
WMem	1	3	2	2	4

Wydajność węzła [bw]:

1:1	w1	w2	w3	w4	w5
bwW	1000	2000	3000	4000	5000

Oczekiwana liczba zapytań [Hz]:

1:1	z1	z2	z3	z4	z5
Hz	5	10	15	20	25

Oczekiwana liczba metod [Hm]:

1:1	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10
Hm	100	95	90	85	80	75	70	65	60	55

liczba losowań poprawnych macierzy: 5

parametry powiązań:

- parametr k1: 2.00
- parametr k2: 3.00
- parametr k3: 3.00
- parametr k4: 4.00

Warunek stopu GA

Liczba generacji: 10

Procent zmian: 0.500

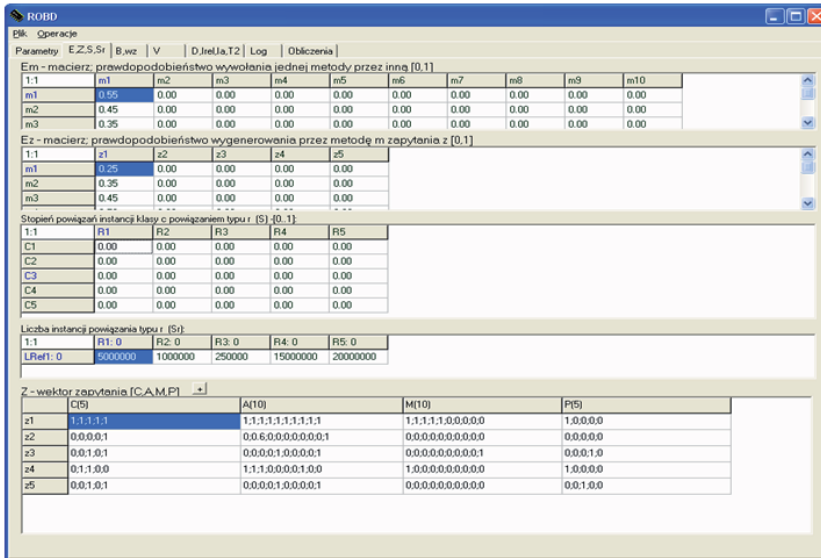
Rys. 13. Ekran interfejsu do konfiguracji komputerowego systemu rozproszonego

Z kolei poniższy rysunek prezentuje ekran z interfejsem do wprowadzania i modyfikacji parametrów konfiguracyjnych podstawowych elementów obiektowej bazy danych (rys. 14).

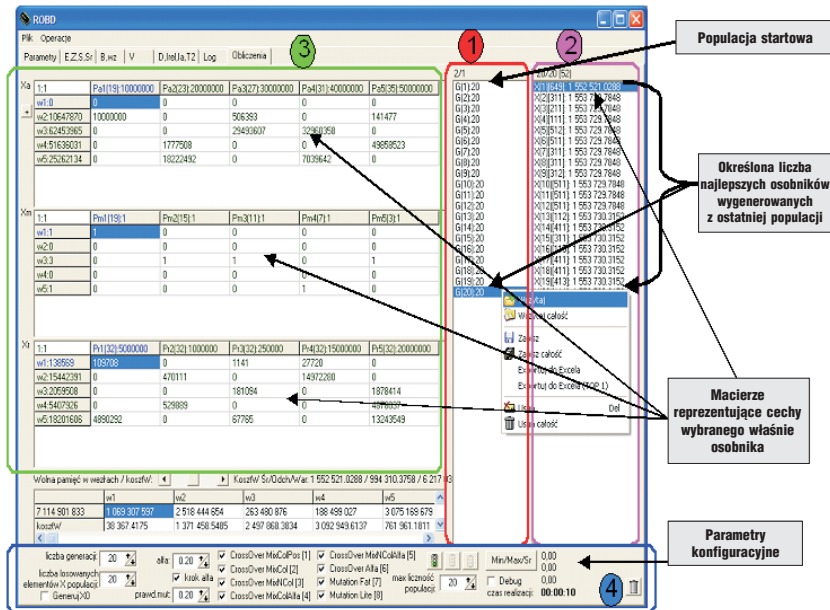
Poniższy rysunek (rys. 15) ilustruje symulacyjne wyznaczenie serii populacji pośrednich i wreszcie końcowej, z której wyodrębnione zostaje — najlepsze z dotąd wyznaczonych rozwiązań — rozmieszczenie zasobów obiektowej bazy danych.

W wyniku rozwiązania problemu można dodatkowo pokazać szereg ciekawych wyników związanych z analizą rozmieszczenia zasobów obiektowej bazy danych. Na poniższym rysunku (rys. 16) prezentowany jest wykres zmian oczekiwanych kosztów jednostkowych (w tym przypadku oczekiwanego jednostkowego czasu odpowiedzi na zapytania) dla poszczególnych węzłów systemu rozproszonego wraz z kolejnymi wygenerowanymi populacjami. Widać, że minimalizacja czasu odpowiedzi systemu bazodanowego prowadzi również do równoważenia obciążenia

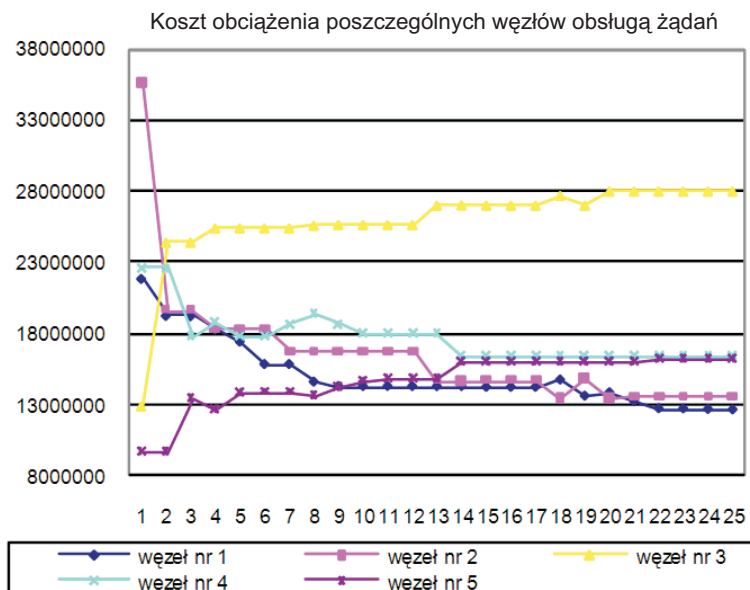
żenia węzłów systemu w zakresie wymuszonym przez ograniczenia określone dla badanego zadania.



Rys. 14. Ekran interfejsu do konfiguracji obiektowej bazy danych



Rys. 15. Wyznaczanie rozmieszczenia elementów obiektowej bazy danych



Rys. 16. Wykres zmian obciążenia węzłów dla kolejnych wygenerowanych populacji

Podsumowanie

Oszacowanie kosztów (w sensie czasu reakcji bazy danych na zapytanie) realizacji zapytań do obiektowej bazy danych jest niezbędne dla wyznaczenia optymalnej alokacji jej zasobów. Okazuje się, że pomimo złożoności modelu matematycznego i stochastycznych strumieni zapytań kierowanych do obiektowej bazy danych, oszacowanie tych kosztów bazy danych jest wykonalne. Oszacowanie kosztów pozwoli na wyznaczenie rozmieszczenia (alokację) zasobów (metod, atrybutów i powiązań) w ROBD przy pomocy efektywnych metod wyznaczania rozwiązań sformułowanego zadania optymalizacji. Jedną z efektywnych metod wyznaczania rozwiązania zadania rozmieszczenia zasobów rozproszonej obiektowej bazy danych jest metoda symulacyjnego wyznaczania rozmieszczenia zasobów w ROBD.

Praca dotyczy jednego z podstawowych aspektów związanych ze zwiększaniem wydajności systemu z obiektową bazą danych — alokacji jej elementów. Metoda przeznaczona jest głównie dla wielkoskalowych rozproszonych obiektowych baz danych. Jest to podejście ilościowe w optymalizacji rozmieszczenia zasobów obiektowych baz danych. Metoda optymalizacji rozproszenia elementów może być stosowana zarówno na etapie projektowania systemu, jak i w czasie jego eksploatacji (przerwy techniczne/konserwacyjne) lub jako narzędzie wspomagające ocenę wariantów projektu tej klasy systemów. Metoda abstrahuje od szczegółów konkretnej architektury systemów rozproszonych.

Artykuł wpłynął do redakcji 14.04.2008 r. Zweryfikowaną wersję po recenzji otrzymano w październiku 2008 r.

LITERATURA

- [1] J. ARABAS, *Wykłady z algorytmów ewolucyjnych*, WNT, Warszawa, 2001.
- [2] D. GOLDBERG, *Algorytmy genetyczne w zastosowaniach*, WNT, Warszawa, 1995.
- [3] B. KORZAN, *Procesy stochastyczne i teoria niezawodności*, cz. 1. *Procesy stochastyczne*, Wydawnictwa WAT, Warszawa, 1986.
- [4] Z. MICHAŁEWICZ, *Algorytmy genetyczne + struktury danych = programy ewolucyjne*, WNT, Warszawa, 1996.
- [5] K. SUBIETA, *Teoria i konstrukcja obiektowych języków zapytań*, Wydawnictwo PJWSTK, Warszawa, 2004.
- [6] K. SUBIETA, *Obiektowość w projektowaniu i bazach danych*, Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1998.

J. KOSZELA, T. NOWICKI

Investigation of the allocation strategy of elements in an objective database

Abstract. In this work, the method of efficiency investigation of a distributed objective data base is presented, particularly in the aspect connected to the evolution of the efficiency of distributed data bases using an allocation of elements. This work contains the definitions of model of classes, objects, attributes, methods and the connections in a distributed object-oriented database.

The model question of the database, the structure of a distributed computer system, distribution model of resources of an object-oriented database have been defined, as well as the response time of the realisation answer system question via a selected node in the computer network including the whole distributed computer system was estimated. The problem of resource allocation in an object-oriented database has been formulated. The method for the allocation of attributes, methods and connections of an object-oriented database is described. The method is connected with well-known genetic algorithms. Several methods for the evolving population, in order to obtain optimal solutions, are presented. Software application for determining the allocation of object-oriented data base elements is also shown. Several characteristics of various solutions obtained are illustrated.

Keywords: information system modelling, objective data bases, distributed data bases, allocation of data base elements

Universal Decimal Classification: 681.3.05